SynFinder: A System for Domain-Based Detection of Synonyms Using WordNet and the Web of Data

Matteo Lombardi and Alessandro Marani[(✉)]

School of Information and Communication Technology, Griffith University,
170 Kessels Road, Nathan, QLD 4111, Australia
{matteo.lombardi,alessandro.marani}@griffithuni.edu.au

Abstract. The detection of synonyms is a challenge that has attracted many contributions for the possible applications in many areas, including Semantic Web and Information Retrieval. An open challenge is to identify synonyms of a term that are appropriate for a specific domain, not just all the synonyms. Moreover, the execution time is critical when handling big data. Therefore, it is needed an algorithm which can perform accurately and fast in detecting domain-appropriate synonyms on-the-fly. This contribution presents SynFinder which uses WordNet and the web of data. Given a term and a domain in input, WordNet is used for the retrieval of all the synonyms of the term. Then, synonyms which do not appear in web pages related to the domain are eliminated. Our experimentation shows a very good accuracy and computation performance of SynFinder, reporting a mean precision of 0.94 and an average execution time lower than 1 s.

Keywords: Synonyms · Semantic Web · Synonyms extraction · Domain-based synonyms detection

1 Introduction

The extraction of synonyms is a current and popular topic in literature for the many possible applications in different areas of Semantic Web (SW), from query expansion to ontology matching [17,20]. In SW, the identification of lexical relationship of terms is a critical task because different words can have the same or similar meaning.

A first approach for the retrieval of synonyms is the usage of traditional dictionaries such as WordNet [10] and Wiktionary[1] among others. WordNet is a well-established English lexical database that provides meaning and synonyms of a term in different contexts. The structure of WordNet is mainly based on the synonym relationship among words. These synonyms are grouped into sets

---

[1] http://www.wiktionary.org accessed on 27-06-2015.

called Synsets formed by words that (i) have the same meaning, and (ii) are interchangeable in different contexts. Presently, WordNet contains more than 110,000 Synsets.

Wiktionary is a free-content multilingual dictionary that, similarly to Wikipedia, allows users to modify translations, definitions, synonyms and other information available in it. It currently offers 4,039,912 entries with English definitions from over 1550 languages[2]. However, as reported in the next section, most of the dictionary-based works for the retrieval of synonyms use WordNet.

Another possible source of synonyms is the web. On one side we have Google Translate that may represent a valid tool for synonyms retrieval using data from the web. Unfortunately, at the time of this study, the APIs of Google Translate[3] offers a service for translation purposes only, with no possibility to retrieve synonyms of terms using such service. On the other side, a part of the research community utilizes web pages for the extraction of synonyms via patterns [20,21]. This approach promises the identification of synonyms as they are actually used on the web, but it cannot be used for real-time synonyms extraction, due to the time that is required to parse the web. Another important consideration is that those contributions do not focus on domain-based detection of synonyms.

Therefore, an interesting challenge is the development of a technique that uses both dictionaries and the web for a proper retrieval of synonyms of a term in a short time. In fact, dictionaries can offer the reliability of a correct set of synonyms, and the web can be used to refine the synonyms according to a domain, current trend of usage and other criteria. In particular, the identification of synonyms of a term appropriate for a specific domain is helpful for the construction of domain ontologies, query expansion process, and any other application of Information Retrieval (IR) and SW techniques where it is worth to have a reduced set of synonyms according to a domain.

In this paper it is proposed a new approach for synonyms detection that is (i) focused on a domain, (ii) performed in a short time to be suitable for real-time applications, and (iii) based on reliable sources of lexical relationships. To achieve those criteria, in this study it is addressed an hybrid solution based on IR methods that uses both a dictionary and the web of data. From a dictionary, for example WordNet, a set of synonyms of a term in different contexts is retrieved, with the assurance of correctness of the retrieved set. Then, such set is reduced to only those synonyms that appear in web pages related to the specific domain. This approach aims to produce quickly a set of synonyms that are appropriate for the domain of interest, instead of all the possible synonyms reported by the dictionary. In this way, the proposed approach may be used by any IR or SW system that is domain based, without a significant impact on the performance of the system in terms of execution time.

---

[2] https://en.wiktionary.org/wiki/Wiktionary:Main_Page accessed on 27-06-2015.
[3] https://cloud.google.com/translate/v2/using_rest accessed on 29-06-2015.

2 Background and Related Works

The problem of detection of synonyms, especially domain-based extraction of synonyms, finds multiple applications in IR and SW as discussed along this section. In order to have a more clear picture of such applications, Table 1 reports an overview of some significant studies where techniques for the extraction of synonyms have been proposed or applied. It is particularly reported the purpose for which synonyms have been useful and what source of lexical relationships is used. Most of the analysed contributions detect synonyms from ready-to-use dictionaries and the most popular is WordNet (refer to Table 1). Other studies try to define patterns for conducting the extraction of new synonyms from the web. These solutions are expected to produce more recent sets of synonyms than current dictionaries like WordNet. However, that process requires time and it is not applicable for real-time synonyms detection applications, which is the focus of this paper.

Table 1. An overview of some contributions which use techniques for synonyms detection.

Paper (Year)	Purpose	Lexical relationship source
[6] (2008)	Ontology matching	WordNet
[19] (2010)	Relationships among concepts	WordNet
[5] (2011)	Ontology matching	Wiktionary
[12] (2011)	Domain ontologies creation	WordNet
[13] (2012)	Extraction of key concepts from ontologies	WordNet
[9] (2013)	Creation of a reverse dictionary	WordNet
[15] (2013)	Query expansion	WordNet and Linked Data
[1] (2013)	Sentiment analysis	WordNet
[11] (2013)	Textual entailment recognition	WordNet
[2] (2014)	Most frequent sense of a word	WordNet
[21] (2014)	Synonyms extraction	Corpus-driven
[3] (2014)	Query generation and expansion	Query logs
[16] (2015)	Sentence similarity	WordNet
[7] (2015)	Query expansion	WordNet
[20] (2015)	Query generation and expansion	Query logs
[4] (2015)	Synonyms extraction	Web

The English language, as other languages, has many terms that have the same or similar meaning. For this reason, the refining of the set of synonyms for a domain is a critical task for the improvement of the retrieval phase of IR systems and ontology management in SW.

For example, a user query has to be well expanded in order to effectively retrieve all the items that meet the query. Many techniques have been proposed

for query expansion based on users' characteristics, web navigation history and background knowledge among others [3,7,8,18,20], some of them using synonyms of terms [3,7,20]. In fact, once the domain of interest of the user has been deducted, the issue is to identify key terms for the query expansion process, including synonyms. Also the construction process of domain ontologies can benefit of a real-time domain-based synonyms detection [12].

As Table 1 shows, the most current popular techniques for synonyms detection are based on: query logs, web and dictionaries. The query logs of users are mostly used when it is conducted a query expansion or generation. For such task, the query logs allow to have a set of alternative words that have been used by users in the past to formulate queries about a topic. Instead, the web is mainly involved for studies that aim to improve current dictionaries with the most recent usage of terms and their synonyms. Finally, the most widespread source of lexical relationships of words are the dictionaries, especially WordNet. They are mostly used because they offer reliable relationships in a very short time. WordNet also provides APIs which make an integration in IR and SW systems easy. More interesting, such dictionary has been involved in several studies about ontologies, from domain ontologies creation to extraction of key concepts from ontologies [6,12,13].
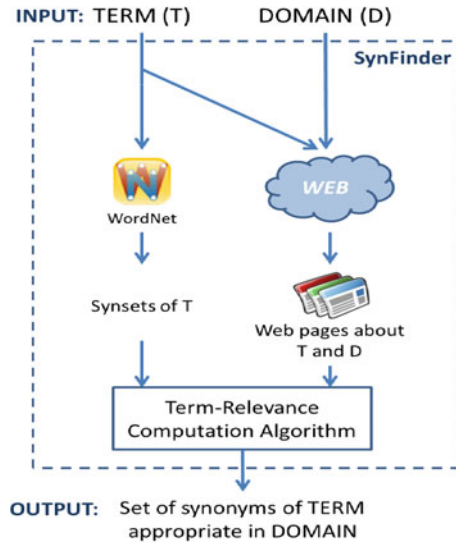

Fig. 1. The structure of SynFinder.

However, to the best of our knowledge, no studies about domain-based detection of synonyms have been proposed. Therefore, in this study it is suggested a novel approach for domain-appropriate synonyms detection which promises a highly reliable synonyms identification in a short time. As current studies use WordNet and other dictionaries to get synonyms quickly but correctly, those

criteria have been the guidelines for the design and development of the system proposed in this contribution, called SynFinder. It combines the reliability of WordNet with the web for the computation of synonyms relevance in a domain. The main goal is to perform such task with high accuracy but low execution or computation time. In this way, the research community can benefit of such system for getting domain-appropriate synonyms in a time that is only a bit longer than just using a present dictionary.

3   Structure of the System

This section reports the main characteristics of SynFinder and discusses a specific configuration of its settings for an effective domain-based deduction of relevant synonyms.

Figure 1 shows the structure of SynFinder, where it is possible to identify the most important parts of the system: *Input Data*, *Dictionary*, *Web Dataset*, *Term-Relevance computation algorithm* and *Output Data*. These parts of SynFinder are described in the following paragraphs.

Input data. The data in input to SynFinder are two strings: the term (T), and the domain of interest (D).

Dictionary. SynFinder uses a dictionary to get all the synonyms of T in different contexts. For this phase, SynFinder can use any dictionary that offers APIs, like WordNet does.

Web Dataset. SynFinder queries the web through current search engines. In this regard, search engines are used as access points to the huge amount of data in the Internet. In addition, search engines provide a structure and an order to the information retrieved from the web, allowing to select only the top-N results that are closer to a query instead of millions of web pages, without losing valuable information. Having less data to analyze, the speed of the synonym detection process can be significantly improved. To query the search engine, T and D are concatenated and given in input to the search engine for retrieving a set of web pages about the term T in domain D. Again, any search engine can be used to perform this task, and it is even possible to combine the results from different search engines in order to consider different sources of information, as shown in Fig. 2.

Term-Relevance computation algorithm. The computation of domain-appropriate synonyms is mainly performed by the *Term-Relevance computation algorithm*, designed and implemented during this study. Currently, this algorithm considers only title and snippet of the documents in input, because title and snippet report short descriptive information about the content of web pages that is close to the query. It is expected that title and snippet contain the term T and/or its appropriate synonyms in domain D. Most of the current popular search engines

structure the results presenting, among other information, the title of the page, and the snippet composed using parts of the page where some of the keywords in the query, and/or their synonyms, appear. A first analysis of the problem could lead to compute the relevance of a synonym in a set of web pages using the well-known TF-IDF score. Such score is very popular in IR and it is useful to evaluate the relevance of a term in a set of text-based documents [14]. However, for the purpose of this study it is useful to calculate, for each candidate synonym of T, how many documents contain the synonym in the title or the snippet compared to the number of total retrieved documents; in essence, the document frequency of a synonym. Hence, the score used in this algorithm consists only of the Document Frequency df, calculated for each synonym s as follow:

$$df(s) = \frac{\mid PostingsList[s] \mid}{\mid Docs \mid} \tag{1}$$

where $Docs$ is the set of documents, and $PostingsList$ is a dictionary of terms that records the list of documents where the term appears. Such postings list is built prior to the computation of the document frequency of terms considering only the title and snippet of the web pages in $Docs$. An important characteristic of the document frequency is that $df(s) \in [0, 1]$. Using Formula 1 and the documents retrieved from the web, the algorithm computes the domain-relevance of all the synonyms of T coming from the dictionary (WordNet in the case of SynFinder).

Output Data. After the computation of the relevance of each candidate synonym, only the synonyms with relevance higher than 0 are grouped to form the output of SynFinder. As result, the output of SynFinder is a dictionary of domain-relevant synonyms of term T reporting the document frequency of each synonym.

3.1   Parameters of SynFinder

The architecture of SynFinder is formed by different modules, as presented in Fig. 1 and discussed at the beginning of this section. Those modules can work at different settings, so it is possible to choose some parameters for running SynFinder. The most relevant parameters of the proposed system are:

- **Dictionary**: it is a reliable database of lexical relationship of terms. In this study WordNet has been chosen among others due to the established popularity and the offered APIs for a fast retrieval of set of synonyms.
- **Web Dataset**: it is the access point to the web and it has the critical task to retrieve web pages that are related to the term and domain given in input to SynFinder. For this aim, after few tests of SynFinder with different *Web datasets*, we have observed that YAHOO! and Google perform nearly the same, but YAHOO! is surprisingly faster. Hence, YAHOO! has been selected for the implementation of SynFinder using the BOSS Search APIs[4].
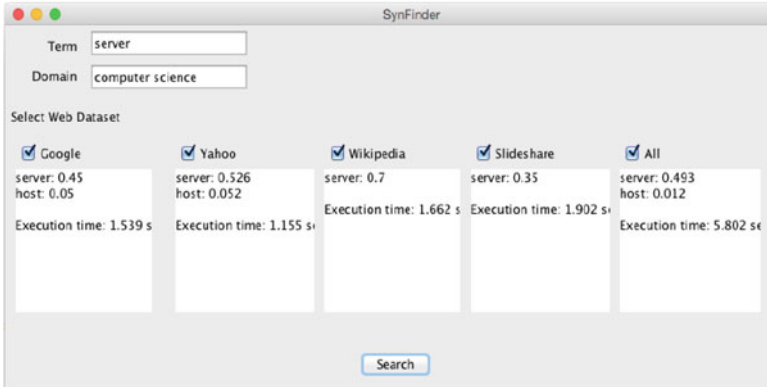
---

[4] https://developer.yahoo.com/boss/search/.

Fig. 2. The *Comparative GUI* of SynFinder.

– **Number of results**: A search engine may retrieve millions of web pages for a query, thus, for a low execution time of SynFinder, only the top 20 web pages are considered. Before to establish such number, the system has been tested with few terms considering the top 10, 20 and 30 web pages of the query results. The top 10 pages are not enough, instead with 20 and 30 top pages SynFinder produces the same set of synonyms, only the document-frequency values are different. Therefore, the configuration with top 20 has been preferred mostly for keeping low the execution time.
– **Features of the results**: Most of current search engines structure the results offering, among other features, title and snippet. These two features are sufficiently good for expressing the content of a web page that is related to a query. So, title and snippet have been selected as the features to be considered by the *Term-Relevance computation algorithm* of SynFinder.
– **Score**: it is used by the *Term-Relevance computation algorithm* to calculate the relevance of the synonyms for the input domain. An appropriate score proposed in this study for this task is the *document frequency* presented earlier in Formula 1.

The configuration of SynFinder presented here is the one used for the experimentation, so more details are reported in Sect. 4.

Every parameter is very important for the execution of SynFinder, but the *Web Dataset* is the most important. Indeed, it determines the quality and significance of the set of web pages that are in input to the *Term-Relevance computation algorithm*. For this reason, SynFinder is presently provided with a *comparative GUI*, showed in Fig. 2, for the comparison of the results that SynFinder produces with different *Web Datasets* (the other parameters are in common).

In particular, the developed GUI offers the possibility to specify the term, domain and web sources. The resulting synonyms are presented in the text areas below the name of the selected search engines. In each text area is also reported (i) the document frequency of the synonyms (in the range *[0,1]*) according to

the web pages retrieved by the respective *Web dataset*, and (ii) the execution time in seconds.

4   Experimentation

The performance evaluation of the proposed system has been conducted with an experiment to test the accuracy and computation time of SynFinder. The experiment has been run on an iMac machine with 2.66 Ghz Intel Core 2duo processor, 4 gigabyte of RAM and OS X Yosemite V.10.10.3.

For this experimentation, WordNet V3.1 has been adopted as dictionary of the system. WordNet has also been used by the authors for the identification of possible different domains of a word, looking at the contexts of meaning proposed by such dictionary for a term. Before running the experiment, we have used the *comparative GUI* presented in Sect. 3.1 for the execution of SynFinder with few terms just for selecting the search engine for this experiment. As result, YAHOO! and Google produced nearly the same results but YAHOO! resulted faster, thus YAHOO! has been selected as search engine for the exploration of the web. In addition, the best results were achieved considering the first 20 web pages retrieved by the search engines. Hence, for this experimentation the following system's parameters have been used:

– Search engine: YAHOO!
– Number of web pages analysed: 20
– Score for term relevance: Document Frequency
– Dictionary: WordNet V3.1

4.1   Methodology

The objective of this experimentation is to report the accuracy and computation performance of SynFinder. Both performance evaluations of the system have been conducted with a test set of domain-related synonyms of terms manually defined by the authors themselves. Each element of the test set is a triple made by term, domain and the set of synonyms as retrieved from the dictionary. In addition, each synonym has a flag that says if it is appropriate or not for the domain as decided by the authors. The appropriateness of a synonym has been decided considering the *sense* of the set of synonyms as retrieved from WordNet. For example, the term 'array' has synonyms 'raiment' and 'regalia' for the *sense* 'especially fine or decorative clothing'. Moreover, from a *sense* it is possible to deduct the domain, in that case 'clothing'. Hence, the synonyms 'raiment' and 'regalia' are appropriate for the term 'array' in domain 'clothing'. With such approach, 32 triples have been produced and used as test set for this experimentation, a small sample is reported in Table 2. For the evaluation of the accuracy performance of the system, it has been adopted the cosine similarity, precision and recall which are common accuracy measures of IR systems [14]. For each entry of the test set, such measures have been calculated for comparing the set of

Table 2. A sample of the test set reporting the term, domain and a list of synonyms as retrieved by WordNet with their domain-relevance flag (0 non-relevant, 1 relevant).

Term	Domain	Set of synonyms
server	computer science	server:1;waiter:0;host:1
server	hospitality	server:1;waiter:1;host:0
array	social science	array:1;raiment:0;regalia:0;range:0;lay out:0;set out:0;align:1
array	clothing	array:1;raiment:1;regalia:1;range:0;lay out:0;set out:0;align:0
tumor	medicine	tumor:1;tumour:1;neoplasm:1
term	architecture	term:1;condition:0;full term:0;terminus:1;terminal figure:1
term	law	term:1;condition:1;full term:0;terminus:0;terminal figure:0

relevant synonyms produced by SynFinder against the correct ones as stated in the test set. In this experimentation all the synonyms with document frequency greater than 0 have been assigned the value 1. The reason is that if a synonym occurs even once in a very small piece of information of the web (only title and snippet of 20 web pages), then the synonym is likely appropriate for the domain. The cosine similarity has been used to compute the similarity between the vector of relevant synonyms of the test set and the relevance values produced by the system. In practice, the similarity is calculated between two vectors with only zeros and ones. The formula of the cosine similarity between vector A and B is the following:

$$CosineSimilarity(A, B) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i * B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} * \sqrt{\sum_{i=1}^{n}(B_i)^2}} \tag{2}$$

Precision and recall are here useful for further insights of the quality of the results produced by SynFinder. Precision shows the actual relevance of the set of synonyms suggested by the system, recall depicts the capability of the system in retrieving all the relevant synonyms, defined as follow:

$$Precision = \frac{\#true\ positives}{\#true\ positives + \#false\ positives} \tag{3}$$

$$Recall = \frac{\#true\ positives}{\#true\ positives + \#false\ negatives} \tag{4}$$

Where *true positives* are the synonyms correctly labelled as relevant, *false positives* are those synonyms wrongly labelled as relevant by the system and *false negatives* are relevant synonyms considered non-relevant by SynFinder.

Moreover, for the evaluation of the computation performance, the execution time of each call to the system has been recorded. During the execution of each triple it has been registered the time when the system received the input (T_s) and the time when the output was given back (T_e). The difference between the T_e and T_s is the *completion time* for the detection of domain-appropriate synonyms.

4.2   Results

In this subsection we report the accuracy and computation performance of Syn-Finder recorded during the experimentation. Table 3 shows the minimum, maximum and average values of precision, recall and cosine similarity measures registered during the experimentation. Overall, the system performed well having an average precision of 0.94 and average recall of 0.64. Such high precision means that most of the synonyms retrieved by SynFinder are actually relevant for the domain, and, as expected, a high precision causes a low recall. However, a value of 0.64 for recall says that more than 60 % of relevant synonyms are detected by the system, which is not a low recall at all. In addition, the experimentation was run considering only title and snippet of the first 20 results presented by YAHOO!. So, more results can be considered and the web page can be fully analysed in order to increase the recall, but it may lead to a lower precision as well as an increase of execution time. For the application of the synonyms detection problem in IR and SW areas, we believe that the precision is more important than the recall, otherwise the proposed system does not make any significant difference than using only WordNet. We also highlight that some terms are not so popular on the web, thus a low recall may just reflect a disuse of some synonyms in current English. For example, for the term *word* in the domain *social science* we registered a precision of 1 but a recall of 0.25 because SynFinder has retrieved only the term itself, leaving out three other relevant synonyms. One of these is *give-and-take* retrieved by WordNet V3.1 as a synonym of *word* when people exchange different views on a topic. However, that exact syntactic structure may result not common on the web language. Also *discussion* is a valid synonym left out by SynFinder, but it might be present in the body of a web page returned by the search engine, thus a full analysis of web pages instead of only title and snippet may detect it. Anyway, a recall value lower than 0.5 occurs only for 7 out of 32 records. Therefore, a mean value of recall equal to 0.64 is not a big issue and it is better to keep it as it is than lowering the precision of the system.

Table 3. Minimum, maximum and average value recorded for precision, recall and cosine similarity.

	Minimum	Maximum	Average
Precision	0.6	1.0	0.94
Recall	0.25	1.0	0.64
Cosine Similarity	0.5	1.0	0.75

About precision, the lowest recorded value of precision is 0.6 for the term *instrument* in *law* and the term *tone* in *art*; only for those two cases we recorded that value of precision.

With more details about the accuracy performance of SynFinder, Figs. 3 and 4 show the accuracy performance of the system recorded for each element

Table 4. Percentage of cases where precision, recall and cosine similarity have been recorded equal or greater than the respective average values.

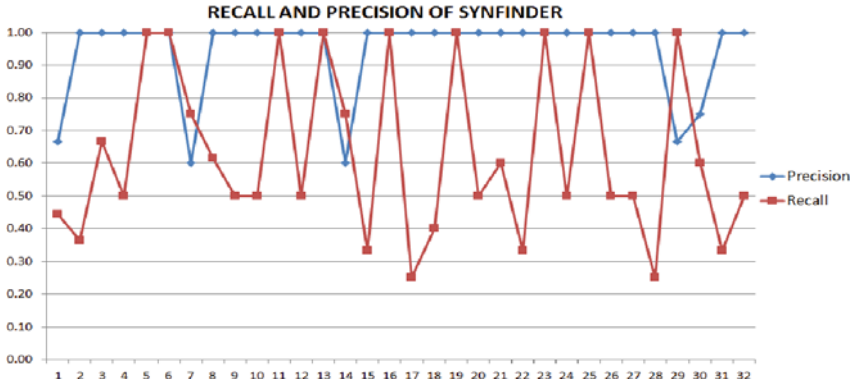|	Precision	Recall	Cosine Similarity
Test cases with measure equal or greater than average	84 %	38 %	38 %

Fig. 3. Accuracy performance of SynFinder in terms of precision and recall.

of the test set. In addition, Table 4 reports another interesting finding about the percentage of entries of the test set for which it has been recorded a value of precision, recall and cosine similarity equal or greater than the respective mean values. Very interesting, for more than 80 % of terms SynFinder detected domain-relevant synonyms with a precision of at least 0.94. From a perspective of computation performance, SynFinder performs well according to the statistics of execution time recorded in this experimentation and reported in Table 5. A very encouraging result is that the longest recorded execution time is less than 2 s, with a very positive average execution time of 0.71 s, very close to the minimum time recorded (0.57 s).

Table 5. Recorded minimum, maximum and average execution time in seconds.

	Minimum	Maximum	Average
Execution time (seconds)	0.57	1.96	0.71

Therefore, at the end of this experimentation we have a very positive analysis of SynFinder performance, showing that it performs very well in terms of both accuracy and execution time, promising a novel valid approach for domain-based detection of synonyms.
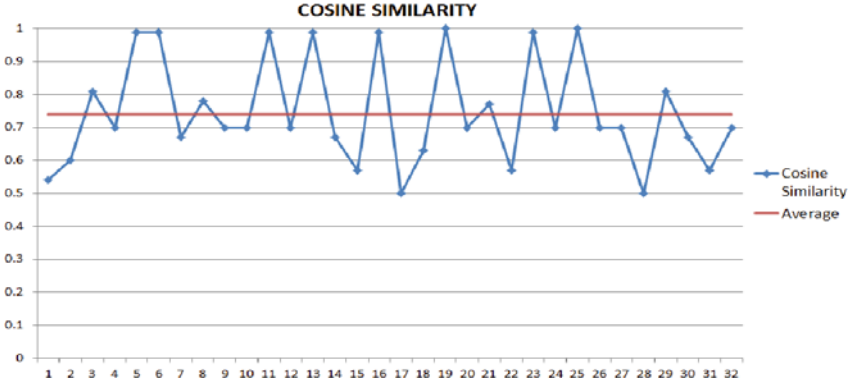
Fig. 4. The cosine similarity between the relevant synonyms as established by Syn-Finder and the actual relevant ones, together with the average value recorded among the 32 cases of the test set.

5   Conclusions and Future Work

At the end of this study, it has been described the SynFinder system and proved its significant effectiveness as a tool for the detection of domain-appropriate synonyms. The combination of a very reliable and popular dictionary, like Word-Net is, with the web has been successful for the proposal of a new and valid approach for the discovery of synonyms of terms suitable for a domain. The experimentation conducted in this study confirms it, with very positive results about both the accuracy and computation performance of SynFinder. Other possible configurations of the system can be evaluated to see whether or not SynFinder significantly benefits of the analysis of entire web pages instead of title and snippet only as proposed in this study. However, the configuration here proposed and analysed performs very well, with a mean precision measure of 0.94 and average execution time lower than 1 s. Therefore, the proposed SynFinder is ready to be integrated or used for systems in SW and IR that would benefit of such a tool.

In this regard, SynFinder should be available online via REST APIs in order to be automatically used by the research community. So, the next step is the deployment of SynFinder as web service for domain-based synonyms retrieval guaranteeing high accuracy of the results and low response time, as reported by the experimentation. Currently, to the best of our knowledge, no other online system offers a domain-based synonyms detection through APIs, neither Google Translate that has APIs for translation purposes only.

In conclusion, SynFinder represents a novel contribution in the field of SW for improving and speeding up the detection of domain-relevant synonyms using WordNet and the web of data.

References

1. Batool, R., Khattak, A.M., Maqbool, J., Lee, S.: Precise tweet classification and sentiment analysis. In: IEEE/ACIS 12th International Conference on Computer and Information Science (ICIS), pp. 461–466. IEEE (2013)
2. Calvo, H., Gelbukh, A.: Finding the most frequent sense of a word by the length of its definition. In: Gelbukh, A., Espinoza, F.C., Galicia-Haro, S.N. (eds.) MICAI 2014, Part I. LNCS, vol. 8856, pp. 1–8. Springer, Heidelberg (2014)
3. Cheng, T., Chakrabarti, K., Chaudhuri, S., Xin, D.: Robust discovery of entity synonyms using query logs, 3 June 2014. US Patent 8,745,019
4. Hu, F., Shao, Z., Ruan, T.: Self-supervised synonym extraction from the web. J. Inf. Sci. Eng. **31**, 1133–1148 (2015)
5. Lin, F., Krizhanovsky, A.: Multilingual ontology matching based on wiktionary data accessible via sparql endpoint (2011). arXiv preprint arXiv:1109.0732
6. Lin, F., Sandkuhl, K.: A survey of exploiting wordnet in ontology matching. In: Bramer, M. (ed.) Artificial Intelligence in Theory and Practice II, pp. 341–350. Springer, US (2008)
7. Lu, M., Sun, X., Wang, S., Lo, D., Duan, Y.: Query expansion via wordnet for effective code search. In: IEEE 22nd International Conference on Software Analysis, Evolution and Reengineering (SANER), pp. 545–549. IEEE (2015)
8. Makvana, K., Shah, P.: A novel approach to personalize web search through user profiling and query reformulation. In: International Conference on Data Mining and Intelligent Computing (ICDMIC), pp. 1–10. IEEE (2014)
9. Méndez, O., Calvo, H., Moreno-Armendáriz, M.A.: A reverse dictionary based on semantic analysis using wordnet. In: Castro, F., Gelbukh, A., González, M. (eds.) MICAI 2013, Part I. LNCS, vol. 8265, pp. 275–285. Springer, Heidelberg (2013)
10. Miller, G.A.: Wordnet: a lexical database for english. Commun. ACM **38**(11), 39–41 (1995)
11. Pakray, P., Neogi, S., Bandyopadhyay, S., Gelbukh, A.: Recognizing textual entailment in non-english text via automatic translation into english. In: Batyrshin, I., Mendoza, M.G. (eds.) MICAI 2012, Part II. LNCS, vol. 7630, pp. 26–35. Springer, Heidelberg (2013)
12. Pariente Juarez, S., Esquivel, H.E., Rebollar, A.M., Suárez-Figueroa, M.C.: CreaDO–A methodology to create domain ontologies using parameter-based ontology merging techniques. In: 10th Mexican International Conference on Artificial Intelligence (MICAI), pp. 23–28. IEEE (2011)
13. Rospocher, M., Tonelli, S., Serafini, L., Pianta, E.: Key-concept extraction for ontology engineering. In: ten Teije, A., Völker, J., Handschuh, S., Stuckenschmidt, H., d'Acquin, M., Nikolov, A., Aussenac-Gilles, N., Hernandez, N. (eds.) EKAW 2012. LNCS, vol. 7603, pp. 413–416. Springer, Heidelberg (2012)
14. Shani, G., Gunawardana, A.: Evaluating recommendation systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 257–297. Springer, US (2011)
15. Shekarpour, S., Hoffner, K., Lehmann, J., Auer, S.: Keyword query expansion on linked data using linguistic and semantic features. In: IEEE Seventh International Conference on Semantic Computing (ICSC), pp. 191–197. IEEE (2013)
16. Shin, Y., Ahn, Y., Kim, H., Lee, S.-G.: Exploiting synonymy to measure semantic similarity of sentences. In: Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication, p. 40. ACM (2015)

17. Shvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges. IEEE Trans. Knowl. Data Eng. **25**(1), 158–176 (2013)
18. Sieg, A., Mobasher, B., Burke, R.: Web search personalization with ontological user profiles. In: Proceedings of the Sixteenth ACM conference on Information and Knowledge Management, pp. 525–534. ACM (2007)
19. Sorrentino, S., Bergamaschi, S., Gawinecki, M., Po, L.: Schema label normalization for improving schema matching. Data Knowl. Eng. **69**(12), 1254–1273 (2010)
20. Tannebaum, W., Rauber, A.: *PatNet*: a lexical database for the patent domain. In: Hanbury, A., Kazai, G., Rauber, A., Fuhr, N. (eds.) ECIR 2015. LNCS, vol. 9022, pp. 550–555. Springer, Heidelberg (2015)
21. Yıldız, T., Yıldırım, S., Diri, B.: An integrated approach to automatic synonym detection in Turkish corpus. In: Przepiórkowski, A., Ogrodniczuk, M. (eds.) Advances in Natural Language Processing. LNCS, vol. 8686, pp. 116–127. Springer, Switzerland (2014)